




第2章 软件架构的概念



目 录

- 2.1 概述
- 2.2 组成派的主要定义
- 2.3 决策派的主要定义
- 2.4 其他定义
- 2.5 参考定义框架

2.1 概述

- 软件架构的定义一出现就存在比较大的争论。
 - 研究人员一般认为：软件架构就是一个系统的草图。
 - 业界人士虽然也认同研究人员对软件架构概念的描述，但实际上很多时候很难用某个软件架构定义来对问题进行系统的、全面的、准确的刻画，也很难解决到满意的程度。
 - 由于学术界和工业界的联系不紧密，导致对软件架构的认识不一致，使得软件架构至今很难有个统一的定义。

2.1 概述

- 软件架构的定义驳杂多端，其中影响较大的定义派别是：**组成派和决策派**。
- **组成派**关注于软件本身，将软件架构看做构件和交互的集合；
- **决策派**关注于软件架构中的实体（人），将软件架构视为一系列重要设计决策的集合。

2.2 组成派的主要定义

组成派定义依据：软件架构主要反映系统由哪些部分组成，以及这些部分是如何组成的，强调软件系统的整体结构和配置。

2.2 组成派的主要定义 (2)

- 1992年 Dewayne和Alexander软件架构定义
 - 软件是由架构元素 (elements)、架构形式 (form) 和架构原理 (rationale) 组成的集合。
 - 软件架构={元素, 组成, 原理},
 - 其中, 元素是指具有一定形式的结构化元素。包括处理元素 (processing elements)、数据元素 (data elements) 和连接元素 (connecting elements)。
 - 架构组成由加权的属性 (weighted properties) 和关系 (weighted relationships) 构成。属性用来约束架构元素的选择, 关系用来约束架构元素的放置 (placement)。
 - 架构原理捕获在选择架构风格、架构元素和架构形式的选择动机。

2.2 组成派的主要定义 (3)

- 1993年 David和Mary定义
 - 软件架构包括组件（**component**）、连接件（**connector**）和约束（**constraint**）三大要素。
 - 组件可以是一组代码，也可以是独立的程序；连接件可以是过程调用、管道和消息等，用于表示组件之间的相互关系；约束一般为组件连接时的条件。
- 1994年Jones认为软件架构是构件以及构件之间的交互规则的集合。
-
- 2011年ISO/IEC/IEEE标准中定义软件架构是某一系统的基本组织结构，其内容包括软件构件，构件间的联系，构件与其环境间的联系，以及指导上述内容设计与演化的原理。

2.3 决策派的主要定义

决策派定义依据：软件架构设计是软件设计的一部分，软件设计实际上是开发人员意志和决策在软件开发过程中的体现，软件架构更是高层领导和架构师意志和决策的体现，强调的是设计决策，所以更加注重架构风格和模式的选择。

2.3 决策派的主要定义 (2)

- 1999年 Booch等
 - 软件架构是一系列重要决策的集合，这些决策关于：软件系统的组织；组成系统的结构元素和它们之间的接口，以及当这些元素相互协作时所体现的行为；如何组合这些元素，使它们逐渐合成为更大的子系统；架构风格；这些元素以及它们的接口、协作和组合。
- 2005年 Anton等
 - 软件架构是架构层次上所有设计决策的集合体，这些设计决策与以下内容有关：架构改造的影响、原理、设计准则、设计约束以及附加需求。
- 2006年 Kruchten等
 - 设计决策+设计，这里的设计指的是设计决策的推理过程。

2.4 其他定义

- Vivek Khare认为软件架构是设计和构建软件应用的科学和艺术，这些软件应用满足生命周期中用户的各种需求；
- Aakash Ahmad认为软件架构是包含设计、演化、构建配置的推理和构件互连关系的高层抽象结构；
- Andreas Rausch认为软件架构是一个针对软件改变的框架；
- Muthu Rajagopal认为软件架构是能够有效组合在一起的软件和硬件构件的集合，这些构件组合后能满足预期需求。

2.5 参考定义框架（I）

- 基于国内外普遍认可的看法推荐如下的参考定义框架：
 - 软件架构包括构件、连接件和约束三种基本元素，以及端口和角色两种元素。
 - 构件作为一个封装的实体，仅通过其接口结构与外部环境交互，而构件的接口由一组端口组成，每个端口表示了构件和和外部环境的交互点。
 - 连接件的接口由一组角色组成，每个角色定义了相关交互的参与者。

2.5 参考定义框架（2）

- 软件架构一般由如下五种元素构成：组件（**component**），连接件（**connector**），配置（**configuration**），端口（**port**）和角色（**role**）。如下图所示。

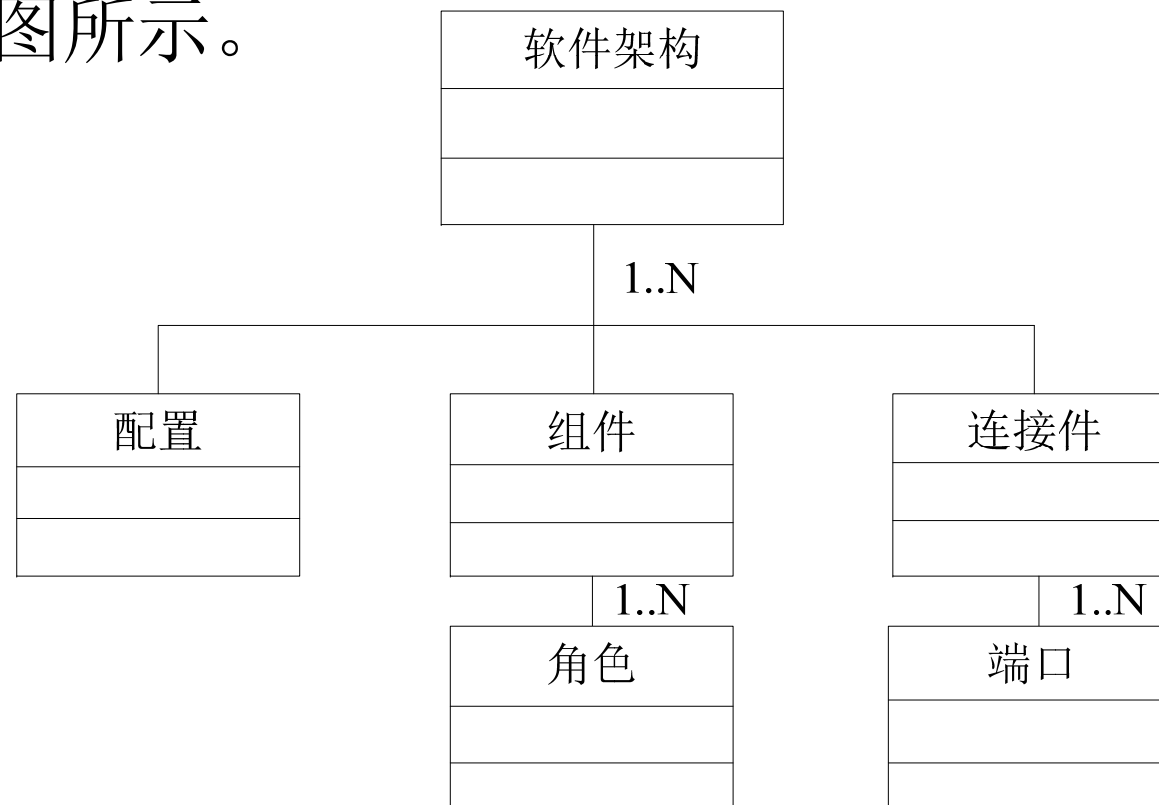


图2.1 软件架构的基本概念

2.5 参考定义框架 (3)

- **组件：** 具有某种功能的可重用的软件模块单元，表示了系统中主要的计算单元和数据存储。组件有两种：复合组件和原子组件。复合组件由其它复合组件和原子组件通过连接而成。
- **连接件：** 表示了组件之间的交互。
 - 简单的连接件有：管道（**pipe**）、过程调用（**procedure-call**）、事件广播（**event broadcast**）等。
 - 复杂的连接件有：客户—服务器（**client-server**）通信协议，数据库和应用之间**SQL**连接等。
- **配置：** 表示了组件和连接件的拓扑逻辑和约束（**constraint**）。

2.5 参考定义框架（4）

- 端口：组件作为一个封装的实体，只能通过其接口与外部交互，组件的接口由一组端口组成，每个端口表示了组件和外部环境的交汇点。
 - 通过不同的端口类型，一个组件可以提供多重接口。
 - 端口可以很简单，如过程调用；也可以很复杂，如通信协议。

2.5 参考定义框架（4）

- 角色：连接件作为建模软件体系结构的主要实体，同样也有接口，连接件的接口由一组角色组成，连接件的每个角色定义了该连接件表示的交互的参与者。
 - 二元连接件有两个角色，如：RPC的角色为 caller 和 callee， pipe 的角色是 reading 和 writing，消息传递的角色是 sender 和 receiver 等。
 - 有的连接件有多于两个的角色，如事件广播有一个事件发布者角色和任意多个事件接收者角色。

作业

- 1. 软件架构决策派定义和组成派定义的本质区别是什么？
- 2. 软件架构与软件系统所处的应用领域有关，说说你对这个问题的理解。